

Adaptive complementary filtering algorithm for mobile robot localization

Armando Alves Neto*, Douglas Guimarães Macharet,
V ctor Costa da Silva Campos, Mario Fernando Montenegro Campos

Computer Vision and Robotic Laboratory – VeRLab, Department of Computer Science,
Federal University of Minas Gerais – UFMG, Belo Horizonte, MG, Brazil

Received: June 30, 2009; Accepted: August 27, 2009

Abstract: As a mobile robot navigates through an indoor environment, the condition of the floor is of low (or no) relevance to its decisions. In an outdoor environment, however, terrain characteristics play a major role on the robot’s motion. Without an adequate assessment of terrain conditions and irregularities, the robot will be prone to major failures, since the environment conditions may greatly vary. As such, it may assume any orientation about the three axes of its reference frame, which leads to a full six degrees of freedom configuration. The added three degrees of freedom have a major bearing on position and velocity estimation due to higher time complexity of classical techniques such as Kalman filters and particle filters. This article presents an algorithm for localization of mobile robots based on the complementary filtering technique to estimate the localization and orientation, through the fusion of data from IMU, GPS and compass. The main advantages are the low complexity of implementation and the high quality of the results for the case of navigation in outdoor environments (uneven terrain). The results obtained through this system are compared positively with those obtained using more complex and time consuming classic techniques.

Keywords: *complementary filtering, localization, outdoor navigation, mobile robots.*

1. Introduction

Recent advances in the computational foundations of robotics have led to the development of new algorithms and efficient methodologies for tackling several fundamental problems in robotics. The sustained improvement in processor performance and sensors technology have made it possible to build vehicles that are substantially autonomous and capable to accomplish a wide variety of tasks in real world. Much of the effort that has been invested to increase the ability of mobile robots to make their own decisions has gone into tackling the core issues of mobility, namely localization and mapping, which are heavily dependent on data interpretation and fusion. Localization consists in estimating the mobile robot’s position and orientation with respect to a given inertial frame, is considered a very hard problem due to the inherent uncertainty and non determinism of the real world. Mapping is a related problem, which may be described as the task of building up a representation of the world based on sensor information.

While moving in an indoor environment, most of the time only the robot’s pose on the plane needs to be known. The pose may be specified by a configuration vector $\vec{P}(t) = [x(t) \ y(t) \ \psi(t)]^T$, in which x and y represent the position of the robot’s center of gravity and ψ represents its bearing angle with respect to some inertial reference frame. The simplifying assumption in the general case is that the terrain traversed by the robot is horizontal and flat, and usually

pavement characteristics such as surface types, and other and anomalies are neglected. This, however, is not the case in outdoor environments, where the terrain’s topography and surface types have a major bearing on the robots displacement and navigation, and should be carefully considered

This paper describes the use of a powerful, yet simple, precise and efficient technique known as CF^{10,5} to tackle the localization problem for mobile robots navigating on uneven terrains in outdoor environments. For our discussion, we cluster the typical irregularities found in outdoor environments into three classes:

- Depressions and bumps;
- Slopes; and
- Different types of surface (e.g. gravel, sand, asphalt, grass).

In such environments, the robot will no longer be moving on a flat horizontal surface, but its attitude will be constantly changing due to terrain irregularities, and hence its pose will have six degrees of freedom, which is represented by the vector $\vec{P}(t) = [x(t) \ y(t) \ z(t) \ \phi(t) \ \theta(t) \ \psi(t)]^T$, where the extra three degrees of freedom are z is the altitude, θ and ϕ are the pitch and the roll angles, respectively. This can also be represented by $\vec{P}(t) = [\vec{x}(t)^T \ \vec{\Phi}(t)^T]^T$, where \vec{x} is the position vector and $\vec{\Phi}$ the orientation vector of the robot.

*e-mail: aaneto@dcc.ufmg.br

Figure 1 exemplifies the importance of having the full attitude information. It is shown the exact moment when the robot transverses an obstacle, with a variation on its pitch angle. Without knowledge of this information the robot is more prone to failure, for example, navigating through a dangerous terrain without realizing it.

Several approaches have been reported in the literature, and in the next section the key ones will be presented and discussed.

2. Related Works

Several approaches for localization have been proposed in the robotics literature, and odometry¹ based techniques have been long used for that purpose. However, obtaining precise measurements from odometric sensors is very hard, for it is plagued with both sistematic and non-sistematic errors. In spite of all these drawbacks, odometry is widely used for navigation on planar, low slippage surfaces, which are prototypical of the majority of two dimensional indoor environments. Even in such controlled scenarios, odometric information alone becomes extremely limited if three-dimensional position and orientation need to be estimated, and this is typically the case in outdoor environments, where wheel slippage and attitude variations are more likely to happen.

One possible approach to overcome the problems present odometric measurements is to include other sensor modalities in the system, and combine all the acquired data in order to reduce uncertainties and thus to improve localization. This new set of sensors may or may not provide redundant information as far as the unknown variables are concerned. In general, one expects that as the number of sensors used to measure (directly or indirectly) a given variable increases, the confidence on the estimation increases.

Another type of sensor that has been widely used in robotic navigation and localization, mainly due to the reduction in cost of MEMS transducers, is the Inertial Measurement Unit IMU. This

I. Odometry is composed of two greek words: hodos (travel) and metron (measure), and consists in using data from actuators to estimate the robot's displacement and its higher derivatives.



Figure 1. Uneven terrain during outdoor environment.

sensor is generally composed of gyros (angular velocity measurement devices) and accelerometers, orthogonally mounted on a base. The data produced by this type of sensor are usually corrupted with noise and bias, which are hard to deal with and demand careful filtering to minimize errors like those due to drift.

The literature brings many different solutions for high level processing and integration of sensor data. One example is described in Ojeda and Borestein¹², where fuzzy logic is used in order to estimate the attitude of a mobile robot equipped with an IMU. However, the experiments presented in the paper showed results only for two dimensional, planar displacements, which disregards problems in extending the odometry information to the three-dimensional space (altitude information), common in systems that implements odometric and inertial data fusion.

In many cases, inertial navigation based systems perform sensor data fusion by means of a KF. Some examples can be found in Liu et al.⁹ and Sukkarieh¹³, where a system composed by an IMU and a GPS is shown. The fusion of data from the sensors was achieved by a standard KF, where system linearity was assumed. The joint use of IMU/GPS is described in Walchko and Mason¹⁶, where an EKF was used, and Li et al.⁸, where a SPKF was employed for data fusion purposes. In both cases, the filters are implementations of the KF for non-linear systems. The main difficulty associated with this method is the complexity in the identification of a good model for sensors and the robotic system. Another drawback is the high computational complexity of the solution, which involves intensive matrix manipulation.

A considerable number of solutions found in the literature is based on techniques for robots equipped with laser range scanners, which simultaneously to navigation, localize itself in the map of the environment it is concurrently building (SLAM). Information obtained from several sensors is then combined and used to correct the estimated position and orientation during navigation.

Müller and Surmann¹¹ present a technique that is used by a large number of 3D SLAM works described in the literature, which topically comprises of a laser scanner mounted on the robot facing the direction of displacement. In some implementations, the laser scanner is also able to rotate about its longitudinal axis. During data acquisition process, the robot moves a certain distance and then stops, at which moment, the laser scanner rotates to collect the information of a specific region. The newly acquired data are aligned with the previously collected ones, a process which improves the estimation of the robot's current position. The main problems with this method are the use of an expensive sensor (laser scanner), and the associated computational complexity of the point registration algorithms.

In Thrun et al.¹⁵, the SLAM technique is used for the navigation of a helicopter model, where the computational complexity problem involved in mapping and localization is a key issue. However, one of the drawbacks of the presented solution is that it only works if the helicopter does not pass by the same place more than once, which is not easy to guarantee as far as the robot's motion is concerned.

Alternatively to the use of Kalman Filters for the data fusion, works based on other methodologies are also found in the literature. One such methodology is the CF, which allows the fusion of trustworthy signals with different frequency bandwidths to produce more precise signals in the time domain. This technique was used with success in aerial navigation as seen in Iscold et al.⁶, Baerveldt and Klang¹, Euston et al.³ and Mahony and Hamel¹⁰. The main advantages of CF are the low computational cost, the faster dynamic response and the simplicity of parameter tuning.

3. Methodology

The method presented in this work is divided in two parts. On the first one, acceleration, angular velocity and magnetic orientation sensor signals are combined to generate an attitude estimate for the mobile robot in the three-dimensional space. On the second part, this attitude is used as part of the robot's spatial position estimation, also considering the signals generated by a GPS receiver and the robot velocity odometry. In both parts, we use a methodology based on the CF principles with cut-off frequency adapted in function of the sensors characteristics, in order to generate better estimates of the robot's configuration in the three-dimensional space.

3.1. Adaptive complementary filter algorithm

Before presenting a more detailed explanation of our methodology, we discuss the main idea of the algorithm used in this paper. This algorithm receives as input a set of data given by several sensors in the mobile robot, and provides as output the robot's pose (position and orientation) in the three-dimensional space. As will be seen in the next sections, the technique has a low computational cost, which allows the use of the technique in real time (online, localization determined while the robot navigates).

Firstly, we consider a mobile robot equipped with a set of sensors composed by three orthogonal accelerometers (\vec{a}), three orthogonal gyroscopes (\vec{g}), two orthogonal magnetometers (\vec{m}), an odometric sensor (v_0), and a GPS receiver (\vec{x}_{gps}). The signals provided by these sensors are combined according to the Algorithm 1, which is divided in two fundamental parts.

```

1:  $\vec{\Phi}(0) \leftarrow \text{initialize}(\vec{a}(0), \vec{m}, (0))$ 
2:  $\vec{X}(0) \leftarrow [0 \ 0 \ 0]^T$ 
3: for  $t = 0$  to  $T$  do
4:   {First step: attitude estimation}
5:    $\vec{\Phi}(t) \leftarrow \text{ATTITUDE\_CF}(\vec{a}(t), \vec{m}(t), \vec{g}(t))$ 
6:   {Second step: position estimation}
7:    $\vec{X}(t) \leftarrow \text{POSITION\_CF}(\vec{\Phi}(t), v_0(t), \vec{x}_{\text{gps}}(t))$ 
8:   {Pose composition}
9:    $\vec{P}(t) \leftarrow [\vec{X}(t) \ \vec{\Phi}(t)]$ 
10: end for
11: return  $\vec{P}$ 

```

Algorithm 1. ADAPTIVE_CF (\vec{a} , \vec{m} , \vec{g} , v_0 , \vec{x}_{gps}).

On the first part, we use the \vec{a} , \vec{m} and \vec{g} signals to calculate an estimate of the orientation vector $\vec{\Phi}$ of the robot. This estimate is computed by the ATTITUDE_CF function, which implements an adaptive CF with variable cut-off frequency.

Similarly, on the second part, we use the angles $\vec{\Phi}$ estimated on the previous step, the v_0 value and the GPS signal to generate an estimate of the robot's position \vec{X} . The adaptive CF computed by the POSITION_CF function is used to deal with problems such as sensor uncertainty and GPS signal loss.

This function and the previous one are explained in the following sections, as well as the initialization steps and the final pose vector (\vec{P}). For the sake of completeness, we describe the main characteristics of each sensor used in each step, and use this information when dealing with the desing of the adaptive complementary filters. We also derive the computational complexity of this and some other data fusion techniques.

3.2. Complementary filter for attitude estimation

In this section, we present the three main stages of the first part of the methodology. They encompass the design of the Adaptive CF for attitude estimation. Initially, the output of the accelerometers are used to produce a first estimate of the roll (ϕ) and pitch (θ) angles of the robot. Next, an estimate of the yaw (Ψ) angle is calculated. A second estimate of these three angles is generated through the integration of the signals from the three orthogonally mounted angular velocity sensors. On the last stage, these signals are fused to generate more reliable estimates for the robot's pitch, roll and yaw angles. This fusion is made by means of complementary filters in which the cut-off frequency is adapted from some characteristics of the used sensors. This result is then used to estimate the spatial localization of the robot on a later part of the methodology.

3.2.1. Attitude estimation based on accelerometer measurements

The measurement of the Earth's gravity vector, by means of the three accelerometers that compose the IMU, is used to generate the first estimate of the robot's attitude, considering only pitch (θ) and roll (ϕ) angles. These sensors are mounted along three orthogonal axes. We also assume the existence of a frame fixed to the Earth's plane, usually know as NED frame, as shown in Figure 2.

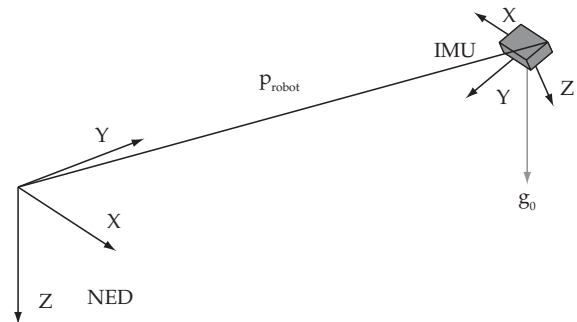


Figure 2. Gravity vector projection (represented in relation to the NED frame) over the IMU axes with three orthogonal accelerometers.

We know from the kinematic principles of rigid bodies that if we consider the NED frame as an inertial frame, the acceleration experienced by a point described by a vector \vec{r}_a in relation to the robot's center of mass (CM), and fixed to the robot's body (e.g. the IMU) is given by $\vec{a}_a = \vec{a}_{CM} + \vec{\alpha} \times \vec{r}_a + \vec{\omega} \times (\vec{\omega} \times \vec{r}_a)$, where \vec{a}_{CM} represents the acceleration of the robot's center of mass, $\vec{\alpha}$ is the angular acceleration of \vec{r}_a , and $\vec{\omega}$ is the angular velocity of this in the space.

If we consider the use of a three axial sensor to measure this vector, we will also observe the effect of the gravity acceleration of the Earth. In general, this effect is subtracted from the final result, in order to simplify the use of the IMU data. Equation 1 represents the values measured by the sensor. In this arrangement, it is possible to represent the gravity acceleration vector as $\vec{g}^{NED} = [0 \ 0 \ g_0]^T$, which is constant in relation to the Earth's frame, and whose value is approximately $g_0 \approx 9.78 \text{ m/s}^2$.

$$\vec{a}_{IMU} = \vec{a}_a - \vec{g}^{NED} \mathbf{R}(\phi, \theta, \psi) + \vec{v}_a. \quad (1)$$

The gravity vector can also be described with respect to the IMU frame by multiplying \vec{g}^{NED} by a rotation matrix \mathbf{R}^{ABC}_{NED} that relates the two frames. However, in order to apply this rotation, it is necessary to compute this matrix or, in other words, to know what are the sensor's orientation angles in space.

Another characteristic of the sensor is the uncertain parameter \vec{v}_a which represents all noise involved in the process. Here, we use a single variable to express gaussian white noise, bias noise and imprecision due to the lack of orthogonality among the sensors.

To simplify the mathematics, it will be assumed that the IMU is mounted exactly on the robot's center of gravity, which in the real case this may not necessarily be true. However, the chosen approach works well even if the difference between the actual location of the center of gravity and the sensor location is not very large. Therefore, we assume that the robot frame, ABC, and IMU frame, are coincident. We will also ignore all types of noise that corrupt the signals measured in this case. These considerations allow us to simplify the previous equation to comply with Equation 2.

$$\vec{a} = \vec{a}_{IMU} = -\vec{g}^{NED} \mathbf{R}(\phi, \theta, \psi). \quad (2)$$

Stripping the above matrix equation and rewriting the angles as functions of the projection of the vector \vec{a}_{IMU} on each one of the IMU axes, it is possible to obtain estimates for ϕ and θ using the measurements from each of the accelerometers, as seen in equations 3 and 4.

$$\theta_a = \arcsin\left(\frac{a_x}{\|\vec{a}\|}\right), \quad (3)$$

$$\phi_a = \arcsin\left(\frac{a_y}{-\|\vec{a}\| \cos \theta_a}\right), \quad (4)$$

where a_x and a_y represent the components of the acceleration vector measured along the IMU X and Y axes.

It is important to note that a few assumptions are made at this stage, by considering the estimated values. First, besides being displaced on the robot's center of gravity, the sensors were mounted along perfectly orthogonal axes. Therefore, the measurements do not present misalignment bias, or if it exists, it is negligible. In addition, there are no other biases on the sensors, and the measurement noise is additive, approximately gaussian with zero mean.

Another important point is that Equation 3 may present singularity problems for $\theta_a = \pm \pi/2$ angles. However, under normal operating conditions, the robot's pitch angle is restricted to values smaller $\pi/2$. Likewise, it is assumed that the roll angle is restricted to even lower values than those of the pitch angle.

An important assumption is that, at each time instant, the accelerometer is measuring only the gravity vector. This assumption is only valid when the robot is at an inertial state. Although restrictive, this assumption is valid for the range of movements performed by ground mobile robots. Therefore, the low frequency signals from the accelerometers provide reliability to the estimate of ϕ and θ . It will be shown later that this information will be combined with the information from the gyros and magnetometers mounted on the robot.

In the case where the robot is not accelerated, $\|\vec{a}\|$ is equal to g_0 . However, this is not true in practice, especially when the robot navigates through uneven environments. This problem will be dealt with the Adaptive CF at the end of this section.

3.2.2. Attitude estimation based on magnetometers measures

As it was previously seen, it is not possible to estimate the robot's orientation relative to the Earth's North by using acceleration sensors alone. Therefore, a sensor that is capable of providing the robot's orientation with respect to the Earth's plane is required. Magnetometers, usually embedded in a digital compass, are the choice sensors for such case.

In the experiments presented in later on, we show the use of a digital compass composed of two orthogonally mounted magnetometers, which is calibrated to measure the orientation of the Earth's magnetic field. In general, only two element sensors are used, mounted in a plane parallel to the Earth's plane.

As done before, we can represent the signal measured by the compass as a three-dimensional vector \vec{m} composed of the sum of the Earth's North orientation (\vec{N}) transformed by a rotation matrix \mathbf{R}^{ABC}_{NED} in relation to the yaw angle (ψ), and a noise vector \vec{v}_m . This can be seen in Equation 5.

$$\vec{m} = \vec{N} \mathbf{R}(\psi) + \vec{v}_m. \quad (5)$$

Again we make some assumptions in order to simplify the problem of estimating the orientation of the robot. The first one, as before, is to neglect the uncertainties involved in the process. Second, we will assume that the plane represented by the two orthogonal magnetometers is always parallel to the Earth's plane.

Equation 6 computes the robot's yaw angle as a function of the magnetometers measurements m_x and m_y in the compass frame. The ψ_m represents a previous estimate of the robot's heading relative to the NED frame, complementing the angular information from the previous section.

$$\psi_m = \arctan\left(\frac{m_y}{m_x}\right). \quad (6)$$

The condition that the compass plane and the earth's plane be parallel, basically constrains this equation to be valid just when parameters m_x and m_y (that are the values measured directly by the magnetometers) are aligned with the Earth's magnetic field. In other words, the compass readings are reliable for small values of roll and pitch. Generally, values larger than 20 degrees begin to induce errors in the attitude value.

At the end of this section, we will show how to use the Adaptive CF methodology to correct problems when the robot has non-null inclination values of ϕ or/and θ .

3.2.3. Attitude estimation based on gyrometers measurements

Besides the accelerometers, the IMU contains three gyroscopes (gyros) that measure the robot's angular velocities about the ABC frame axes. A second estimate for ϕ , θ and ψ can be obtained by integrating the differential kinematic equations that describe the system.

As previously seen, one of the problems is that the mathematics of the attitude parameters using Euler angles representation present singularities in some of its configurations. This can be harmful to the signal integration process, since it produces large discontinuities on the final results. Therefore, we chose to use the unit quaternion representation. The quaternion representation of the vehicle's spatial orientation is given by:

$$\vec{q} = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix},$$

and the pitch, roll and yaw angles are estimated by Equations 7, 8 and 9, respectively.

$$\phi_g = \arctan\left(\frac{2(q_2q_3 + q_0q_1)}{q_0^2 - q_1^2 - q_2^2 + q_3^2}\right), \quad (7)$$

$$\theta_g = \arcsin\left(-2(q_1q_3 + q_0q_2)\right), \quad (8)$$

$$\psi_g = \arctan\left(\frac{2(q_1q_2 + q_0q_3)}{q_0^2 + q_1^2 - q_2^2 - q_3^2}\right). \quad (9)$$

The quaternion values, in turn, are obtained by integrating Equation 10, where g_x , g_y and g_z are the angular velocities corresponding to the roll, pitch and yaw angles, respectively:

$$\dot{\vec{q}} = \frac{1}{2}\Omega\vec{q} \quad (10)$$

where

$$\Omega = \begin{bmatrix} 0 & -g_x & -g_y & -g_z \\ g_x & 0 & g_z & -g_y \\ g_y & -g_z & 0 & g_x \\ g_z & g_y & -g_x & 0 \end{bmatrix}. \quad (11)$$

The initial step of the integration process is determined by the initial attitude estimate obtained by the projection of the gravity vector and the North's heading at time zero, as expressed by:

$$\vec{q}(0) = - \begin{bmatrix} \cos\frac{\phi_0}{2}\cos\frac{\theta_0}{2}\cos\frac{\psi_0}{2} + \sin\frac{\phi_0}{2}\sin\frac{\theta_0}{2}\sin\frac{\psi_0}{2} \\ \sin\frac{\phi_0}{2}\cos\frac{\theta_0}{2}\cos\frac{\psi_0}{2} + \cos\frac{\phi_0}{2}\sin\frac{\theta_0}{2}\sin\frac{\psi_0}{2} \\ \cos\frac{\phi_0}{2}\sin\frac{\theta_0}{2}\cos\frac{\psi_0}{2} + \sin\frac{\phi_0}{2}\cos\frac{\theta_0}{2}\sin\frac{\psi_0}{2} \\ \cos\frac{\phi_0}{2}\cos\frac{\theta_0}{2}\sin\frac{\psi_0}{2} + \sin\frac{\phi_0}{2}\sin\frac{\theta_0}{2}\cos\frac{\psi_0}{2} \end{bmatrix},$$

where ϕ_0 and θ_0 represent the initial values of the roll and pitch angles given by the accelerometers at zero time instant ($\phi_a(0)$ and $\theta_a(0)$), and ψ_0 is the initial angle given by the magnetometers ($\psi_m(0)$).

In this case, the IMU does not need to be close to the vehicle's center of gravity. However the bias of the MEMS gyros can distort the velocity measurements, producing destructive effects on the numerical integration process. The drift caused by this effect produces a low frequency error that hinders the direct use of this process.

On the following development, it is assumed that sensor bias is negligible and that noise is additive gaussian with zero mean. Then, it will be shown that these assumptions are valid, and that the accelerometers, magnetometers and gyroscopes data may be combined by a complementary filtering process.

3.2.4. Attitude data fusion

The output of the previous stages is a pair of estimates of the robot's attitude from the IMU and the compass. As already discussed, the values of ϕ_a , θ_a and ψ_m are reliable as long as the vehicle performs slow movements in horizontally flat terrains. This is the case of a robot moving in an approximately inertial way with small roll and pitch angles. If that is not the case, the parameters ϕ_g , θ_g and ψ_g are susceptible to errors due to the integration of low frequency noise, being therefore, unreliable under larger variations.

Based on the assumptions above, data fusion that considers only the most reliable frequency range of each signal, so as to generate the most reliable result, should be performed. CF is a natural choice, since it is comprised of two digital filters: a low-pass and a high-pass filter, sharing

the same cut-off frequency ω_c . For the experiments presented in the next section, first order digital filters with maximally planar response on the passband (Butterworth) and 10 Hz sampling rate were used. A typical frequency response of the CF filter can be seen in Figure 3.

The first filter significantly attenuates the high frequency accelerations of the first estimate (gravity projection and magnetic yaw orientation). The second filter suppresses the low frequency errors of the second estimate (due to bias and others uncertainties in the integration step of the gyros data). The filtering process is performed by a convolution in the time domain, as shown in Equations 12, 13 and 14. Finally, the two signals are summed in time, generating the final estimate of pitch, roll and yaw angles.

$$\phi = G_l * \phi_a + G_h * \phi_g, \quad (12)$$

$$\theta = G_l * \theta_a + G_h * \theta_g, \quad (13)$$

$$\psi = G_l * \psi_m + G_h * \psi_g. \quad (14)$$

In general, ω_c has a constant value, defined before the filtering process and used directly in the design of the filter. Its value can be different for each variable being filtered, and it is chosen based on the analysis of the frequency spectrum of each signal. It also depends on the sampling time of each sensor.

Here, we propose an adaptive filter in which we allow the filter project for each process iteration. So, we can change the cut-off frequency ω_c in order to give more or less weight to the high and low frequency signals.

In the first step, we choose the initial values for the cut-off frequency as we did for the non-adaptive case, by analyzing the frequency spectrum of "signal" of each angle. These values are defined as k_ϕ , k_θ and k_ψ for the CF of the roll, pitch and yaw angles respectively, and they are considered the ideal values for a non-accelerated robot with small roll and pitch angles.

Next, we consider the problem of calculating the adaptive ω_c for the case of the angles generated by accelerometers. As pointed out before, the estimates of ϕ and θ are valid only if the vehicles is in inertial navigation. In this case, $\|\bar{a}\|$ is equivalent to g_0 , and all assumptions made are correct. Unfortunately, when the robot navigates through irregular terrains, the above assumption is no longer true, and the

angular estimation may fail. So, we need to reduce the cut-off frequency of the CF in order to give less weight to the accelerometers and more weight for the gyros integration.

The Equations 15 and 16 show how this weighting is made. An exponential function was used because of the need to rapidly reduce the ω_c when the measured acceleration vector differs from the gravity vector.

$$\omega_c^\phi = k_\phi e^{-|g_0 - \|\bar{a}\||}, \quad (15)$$

$$\omega_c^\theta = k_\theta e^{-|g_0 - \|\bar{a}\||}. \quad (16)$$

We recall that the compass readings has a notable limitation, in that the heading values are more reliable when ϕ and θ angles of the robot (and consequently, the compass) are close to zero. So, we decide to use the simple solution of estimating the ideal value k_ψ as the maximum absolute value of the angle between these two, as shown in Equation 17.

$$\omega_c^\psi = k_\psi e^{-\max(|\phi|, |\theta|)}. \quad (17)$$

In the end, we have the orientation vector described as:

$$\bar{\Phi} = [\phi \quad \theta \quad \psi].$$

In the next section, we extend this approach in order to estimate the robot's position in the three-dimensional space. We use the same principles of the Adaptive CF to obtain a more reliable fusion between GPS and robot's odometry.

3.3. Complementary filter for position estimation

In this section, we describe the second part of the methodology. A second CF is designed to combine the position estimates given by the robot's odometry sensors and by the GPS sensor, producing a more reliable and more accurate final estimate of the robot's localization on the six-dimensional configuration space.

3.3.1. Position estimation based on GPS measures

On the first stage of this second part, a GPS signal receiver fixed to the robot's body was used to obtain an estimate of its global positioning. Usually, this estimate is given as a function of the latitude, longitude and altitude of the receiver with respect to the Earth's surface.

In this work, however, it is not necessary to know the robot's position relative to the terrestrial inertial reference, since the displacements will be very small compared to the Earth's radius. Hence, it is possible to assume that the Earth is approximately planar throughout the robot's navigation space. A coordinate transformation from latitude, longitude and altitude of the GPS to the differential manifold of the Euclidian tridimensional space coincident with the NED frame is performed, and the GPS output may be considered as the x_g , y_g and z_g signals.

As we will show at the end of this section, it will also be desirable to know the number of satellites N_{sat} seen by the GPS receiver. It will be used to adjust the cut-off frequency

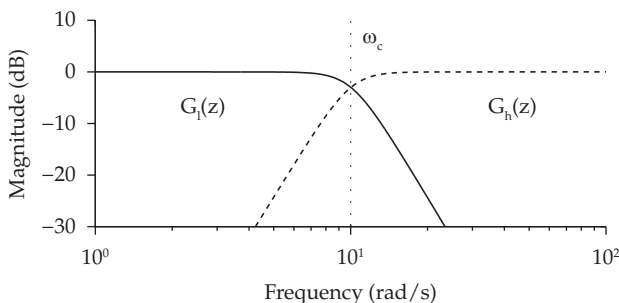


Figure 3. Frequency response of complementary filters with cut-off frequency ω_c .

of the second Adaptive CF to reduce the confidence in this sensor whenever the number of satellites is reduced.

An interesting feature of using the CF technique is that the GPS signal, just as it happens with all sensors, contains noise and other variations. In the specific case of GPSs, one such effect which potentially has a large impact on localization is known as random walk. This effect is caused by high frequency variations which corrupt the satellite signals when the GPS receiver has either stopped or is moving slowly. This, and other similar undesirable effects, can be effectively reduced by filtering out the signal with a low pass filter.

3.3.2. Position estimation based on attitude and odometry

Once the robot's attitude in the tridimensional space is known, the next step consists on estimating the robot's position. One simple solution is to integrate the kinematic differential equation (Equation 18) which describes the robot's movement in space.

$$\begin{bmatrix} \dot{x}_o \\ \dot{y}_o \\ \dot{z}_o \end{bmatrix} = v_o \begin{bmatrix} \cos \psi \cos \theta \\ \sin \psi \cos \theta \\ \sin \theta \end{bmatrix}, \quad (18)$$

where v_o is the measured velocity given by the robot's odometry sensor over time, and ψ and θ are the angles that compose $\vec{\Phi}$, estimated on the first part of our methodology. The integration of this equation generates the x_o , y_o and z_o signals relative to the robot's odometry.

Just as discussed in the previous part, the estimation of a signal by numerical integration generates a measurement that is not reliable due to bias and other sensor noises from sensors. The integration process tends to amplify lower frequency signals of the variable, producing drift over time. Thus, in a complementary way to the GPS signal, the position estimates by odometry may be considered to be more reliable at higher frequencies. The two data streams are then combined using a second CF.

We also consider the fact that the GPS receiver is more affected by noise (such as Random Walk) when it moves with low (or null) velocity. This information will also be used to calculate the adaptive cut-off frequency in order to reduce the confidence on the GPS signal.

3.3.3. Position data fusion

Just as in the first part of the methodology, two different estimates were generated for each one of the variables that we wish to obtain, which present reliable information over different frequency ranges and, therefore, the fusion of these data should be factored in.

For that reason, the Adaptive CF technique was used: two filters G_l and G_h with cut-off frequency ω_c were designed (whose value is not necessarily the same as the previous one). Equations 19, 20 and 21 show the calculation of the final values for the robot's position.

$$x = G_l * x_g + G_h * x_o, \quad (19)$$

$$y = G_l * y_g + G_h * y_o, \quad (20)$$

$$z = G_l * z_g + G_h * z_o. \quad (21)$$

Again, we begin the filters project by choosing an initial value k_p which will be used as the ideal cut-off frequency for all three variable position. In this step we use the same value of ω_c for the CF of x , y and z , as shown in Equation 22.

$$\omega_c^x = \omega_c^y = \omega_c^z = \min(\omega_c^{v_o}, \omega_c^{N_{sat}}), \quad (22)$$

where $\omega_c^{v_o}$ and $\omega_c^{N_{sat}}$ are calculated considering the following constraints.

As mentioned before, the GPS uncertainty increases under low velocities and with a small number of satellites being registered by the receptor. We chose to treat these two effects separately, and take the minimum cut-off frequency between these two constraints.

For the velocity, we use the v_o parameters, provided by the robot's odometry sensor. Equation 23 presents the calculation of the cut-off frequency, where v_{max} represents the maximum translational speed achievable by the mobile robot.

$$\omega_c^{v_o} = k_p \left(1 - e^{-\left| \frac{v_o}{v_{max}} \right|} \right). \quad (23)$$

A similar approach is used in Equation 24 to deal with the number of satellites information. Again, we use the exponential function in order to converge rapidly to low cut-off frequencies when these constraints are unfavorable. Here, the difference is that we have a minimum number of satellites for which this equation is valid, due to the GPS process calculation.

$$\omega_c^{N_{sat}} = \begin{cases} k_p \left(1 - e^{-\left| \frac{N_{sat}}{N_{sat} - N_{max}} \right|} \right), & \text{if } N_{sat} > 3 \\ 0 & \text{elsewhere.} \end{cases} \quad (24)$$

Finally, the vector that represents the robot's position on the configuration space is given by

$$\vec{X} = [x \quad y \quad z]. \quad (25)$$

Along with the orientation vector, this vector is used to compose the pose vector \vec{P} at the end of each iteration of the algorithm. Next, we compare the computational complexity of the algorithm that uses the Adaptive CF, in comparison with the most commons techniques used to make data fusion for robot navigation.

3.4. Computational complexity analysis

In this section we provide a brief analysis of the computational complexity of the Adaptive CF and compared it with other techniques. This is the most important point when

considering the use of this technique for robot localization in the three-dimensional space.

Gaussian Filters, such as Kalman Filter and its derivations (KF, EKF, UKF), are very commonly used in data fusion. The most expensive steps of Kalman Filters are the matrix operations (multiplication, inversion). The asymptotically fastest algorithm known for matrix multiplication is the Coppersmith-Winograd algorithm², with complexity of $O(d^{2.376})$, where d represents the size of the square matrix. In the specific case of Kalman Filters, this size is defined by the number of state variables estimated¹⁴. Here we consider six variables (three positions and three angles).

It is known that the computational cost of the Particle Filter technique depends on the number of particles m that we desire to use. The larger the number of particles, the greater the accuracy of the localization of the robot. According to Thrun¹⁴, the computational complexity of this method is given by $O(m \log m)$. The main problem with this is that the number of particles grows exponentially with the number of states d , which in practice is intractable for three-dimensional space.

The Adaptive CF technique is based on digital filtering theory. The Butterworth filter algorithm has a complexity that is proportional to the order n of the filter, or in other words, it is given by $O(n)$. The proposed methodology uses a filter to estimate each of the variables of the state vector (\vec{P}). It leads to a complexity of $O(nd)$, where d represents the number of variables of the state vector. In the experimental section we will show that first order filters ($n = 1$) are sufficient to build the CF. Therefore, our methodology has a linear complexity. It is less than half of the complexity of frequently used techniques such as UKF.

4. Experiments

The experiments were performed using real data acquired by a mobile robot while traversing an outdoor environment. During the robot's navigation process, it collected and saved all sensor data, which were used as input to the simulation program.

The robot used was a Pioneer 3-AT model designed for outdoor applications. The P3-AT has a good load capacity, and data collection is performed by a laptop mounted on it, which is also responsible for sending commands to the robot. Figure 4 shows the arrangement of sensors on the robot.

In order to compute the attitude and position of the robot, the information provided by an IMU, a GPS, a digital compass, and the robot's odometry sensor were used. The IMU used was the Crista by Cloud Cap Technologies, composed of three mutually orthogonal gyroscopes and three orthogonal accelerometers. A Garmin 18x-5Hz GPS was also used. It provides the absolute values of geographic position of the robot around the world and the number of satellites seen at every moment. The position information was used in terms of the Easting and Northing coordinates (from the coordinate system Universal Transverse Mercator, also called UTM), that are given in meters. Our set of sensors is completed by a KVH

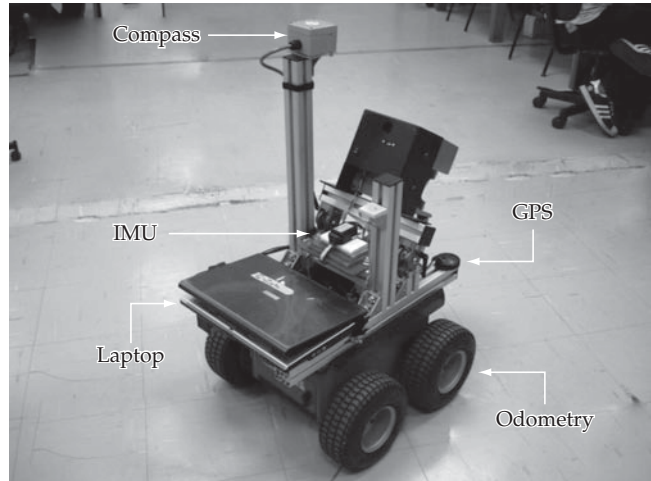


Figure 4. Equipments used in the real experiments.

C-100 Digital Compass, that is responsible for providing absolute values for the orientation of the robot according to the Earth's magnetic north. It is composed by two magnetometers disposed ortogonally in a planar form, and their values can be read directly from a serial interface.

Both IMU and compass present inside filters which are configured to remove noise from its signals. These filters were switch off because they affect the frequency response of the measured variables, and produce great influence on the final result.

As we mentioned before, in order to simplify the calculations, it is assumed that the IMU is mounted exactly at the robot's center of gravity, which in practice may not be necessarily true. Since the difference is very small and the speed of the robot is small, the setup is suitable for our approach. Hence, it is assumed that the robot and the IMU reference frames are coincident.

Three experiments were planned and carried out using the aforementioned setup, and they are describe next.

4.1. Experiment 1

In this first experiment the robot performed a path composed mainly of straight lines, and well defined curves (approximately 90°). The main objectives were to verify the accuracy of the robot's localization, and to verify the possibility of correcting odometric errors by incorporating GPS data.

Figure 5a shows a comparison between three paths: the first one was performed by the robot based on the raw GPS data; the second, computed using our methodology, and the third path estimated by an UKF^{7, 14}. As show in the figure, the raw GPS data is very noisy and clearly provides very low accuracy if used directly. The values obtained using our methodology, however, provided a smoother path closer to the one actually executed by the robot. It is interesting to note that the result was very similar to the one obtained by more complex techniques such as UKF.

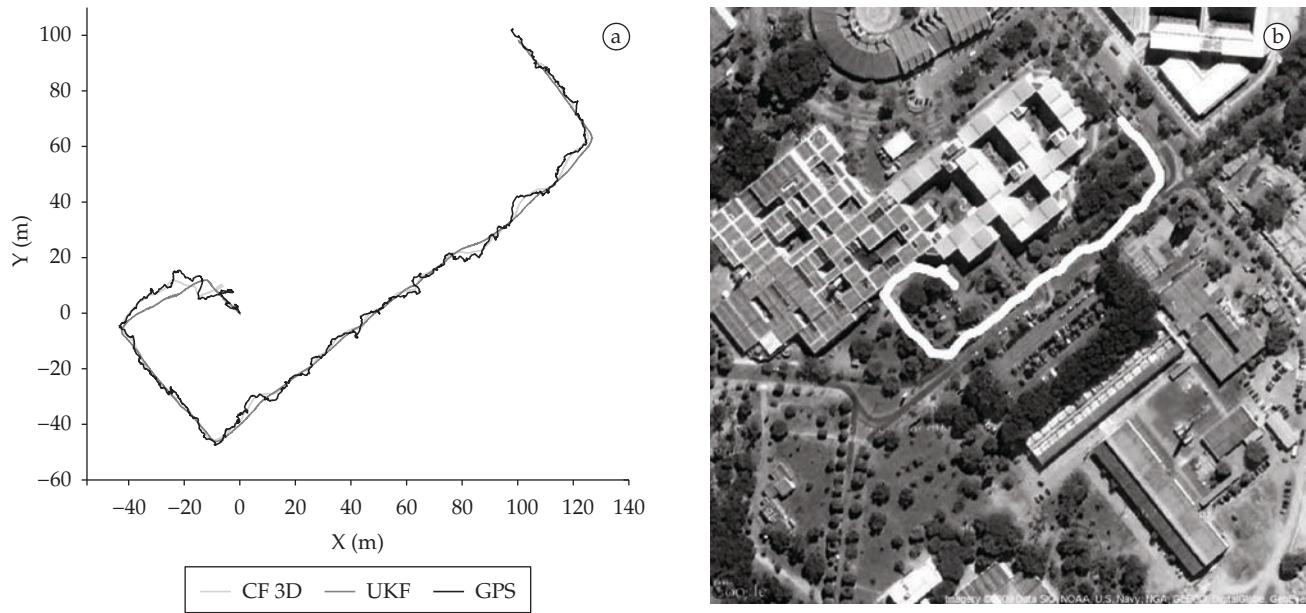


Figure 5. Reconstruction of the path done by the robot on Experiment 1.

The use of the GPS made it possible to correct the velocity errors obtained from odometric measurements. Another benefit of using GPS data is the possibility of obtaining the robot's global localization. A great problem in experiments executed in outdoor environments is to get a ground-truth to compare the results. In order to verify the quality of the results, two different approaches were taken, one to validate the localization and another to check the orientation values.

In order to verify how close the calculated localization is from the actual executed path, the calculated value was drawn on a satellite image using an API⁴. Once the real path is known, it is possible to validate that the calculated value corresponds to the real path. Figure 5b shows the path calculated over an image of the region where the experiments were performed.

The orientation angles were validated by a comparison between two videos. One video corresponds to the executed movements made by the robot, and that were recorded during the experiment. The second video was a 3D animation developed based on the calculated values. It was then possible to conclude that the calculated values for the orientation angles corresponded to the effectively angles obtained by the robot during the execution of the experiment.

As observed in Figure 6, the data still contains a small amount of noise, however, the results were adequate for our purposes, and it represents the actual displacement performed by the robot. Analyzing the standard deviation of the data we have $\sigma_{\phi} = 2.4^\circ$, $\sigma_{\theta} = 2.2^\circ$, while using the UKF the standard deviations are $\sigma_{\phi} = 1.8^\circ$, $\sigma_{\theta} = 1.9^\circ$. We chose to not calculate σ_{ψ} , since the mean is not a good representation of the values obtained.

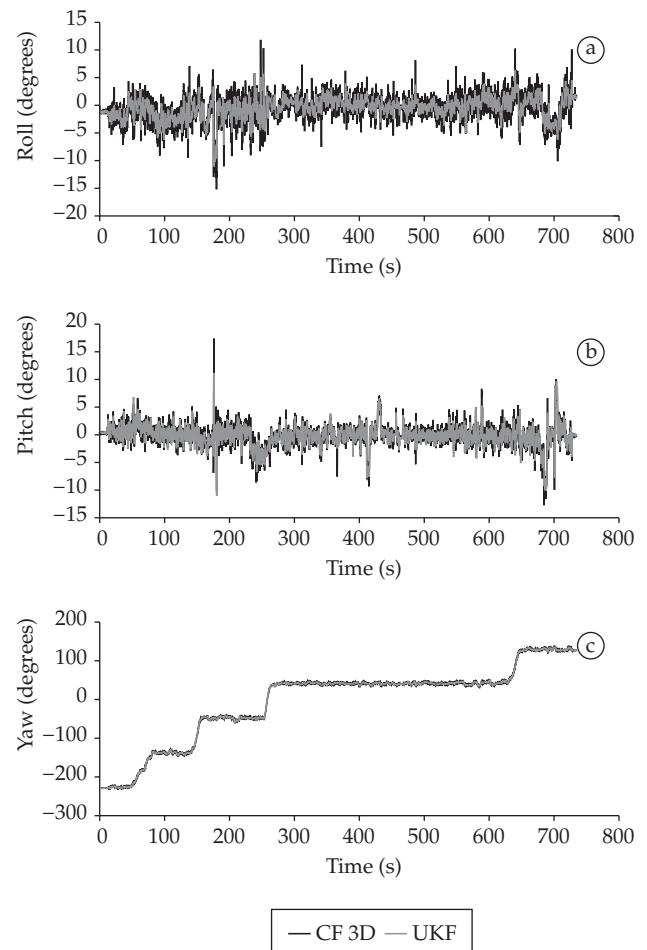


Figure 6. Calculated orientation angles on Experiment 1. a) Roll; b) pitch; and c) yaw.

The angle ϕ (Figure 6a) showed small variations due to imperfections in the terrain and signal noise, but at the end of the experiment it is possible to observe larger variations ($\approx 4^\circ$) when the robot starts to descend a ramp.

The values of θ have, for the most part, zero mean (flat terrain) however, between $t = 230$ s e $t = 270$ s (Figure 6b), the variation of ($\approx 2^\circ$) occurred while it descended an inclined region of the terrain where the experiment was carried out. Three other relevant disturbances, also related to terrain inclination, can be observed in the graph.

The result of ψ can be verified directly, since during the path the robot took four sharp turns (90° curves) that can be clearly seen in Figure 6c, represented by four large steps.

The values of altitude returned from the GPS are fundamental for the computation of the variation in the Z axis. Unfortunately, this value shows a large variation, and as such it is not a good representation of the actual displacement of the robot.

Figure 7 shows a comparison among the altitude calculated by our methodology, the altitude computed by integrating the kinematic equation $z_0(t)$ (Equation 18), and the altitude $z_g(t)$ obtained directly from the GPS. Small variations on the Z axis can be clearly observed at time instants $t \approx 240$ s, $t \approx 410$ s and $t \approx 690$ s. The difference between the values of the highest and the lowest points returned from the GPS is ≈ 25 m, a value which is not consistent with the actual movement performed by the robot, which is well represented by the information obtained from $z_0(t)$.

Thus, the computed values for the variation in altitude were not reliable, and further analysis should be performed for an adequate incorporation of altitude data provided by the GPS. For the moment, and with the technology that was available to us, the measurements provided by the GPS contains much uncertainty. As a matter of fact, a more precise altitude data is required in order to correct the Z using our methodology as well as others, such as UKF.

4.2. Experiment 2

The next experiment consisted on making the robot navigates through different types of surfaces, with the main objective of verify the quality of the adaptation of the filter during the estimation of the orientation angles.

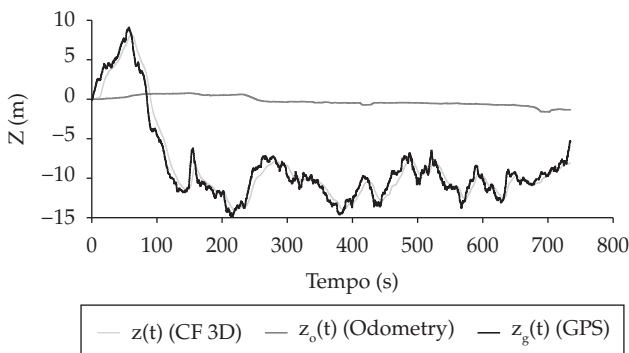


Figure 7. Altitude comparison for the complementary filtering using GPS.

Figure 8 presents all types of surfaces visited the robot during navigation, where Figure 8a presents the most rugged terrain and represents the traversable limit of the used robot (since it moves with great difficulty in this type of terrain), and Figure 8b the less uneven terrain, the other surfaces have both rugged and planar parts.

The images in Figure 9 show the estimated trajectory for the navigation performed by the robot during the experiment.

Figure 10 shows the values of ω_c for each different type of surface during Experiment 2. As it was expected, the rough the terrain the lower the cut-off frequency is, so, on terrain A the cut-off frequency is near zero.

It is possible to conclude that the terrain where the robot navigates directly affects the quality of the estimation, by adding noise to the values of the orientation angles, and consequently the localization.

4.3. Experiment 3

The main objective of Experiment 3 is to verify the quality of the calculation of ω_c in the localization step, based on the number of satellites and the robot's velocity.

It should be noted that during the execution of this experiment the robot never had fewer than seven satellites in sight, so, a complete loss of GPS signal has been simulated.

The images in Figure 11 show the estimated trajectory for the navigation performed by the robot during the experiment.

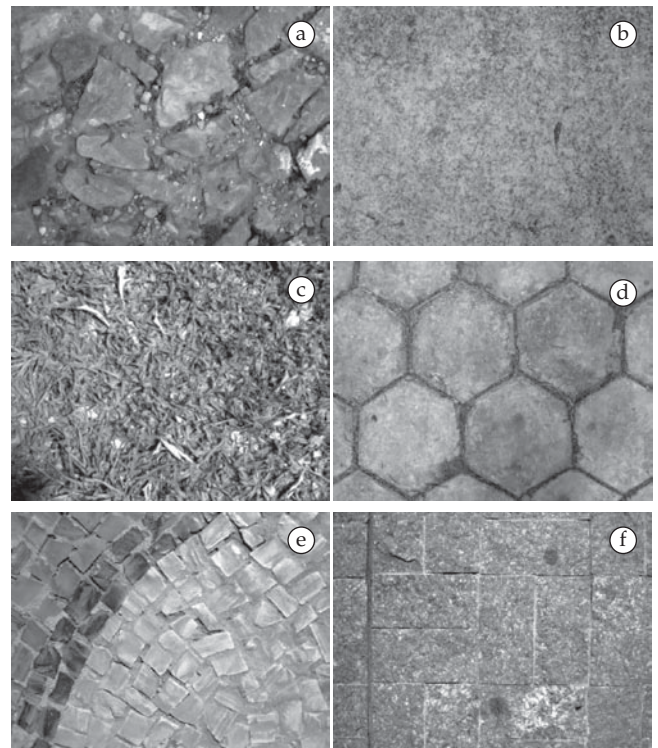


Figure 8. Different types of surface navigated in Experiment 2.

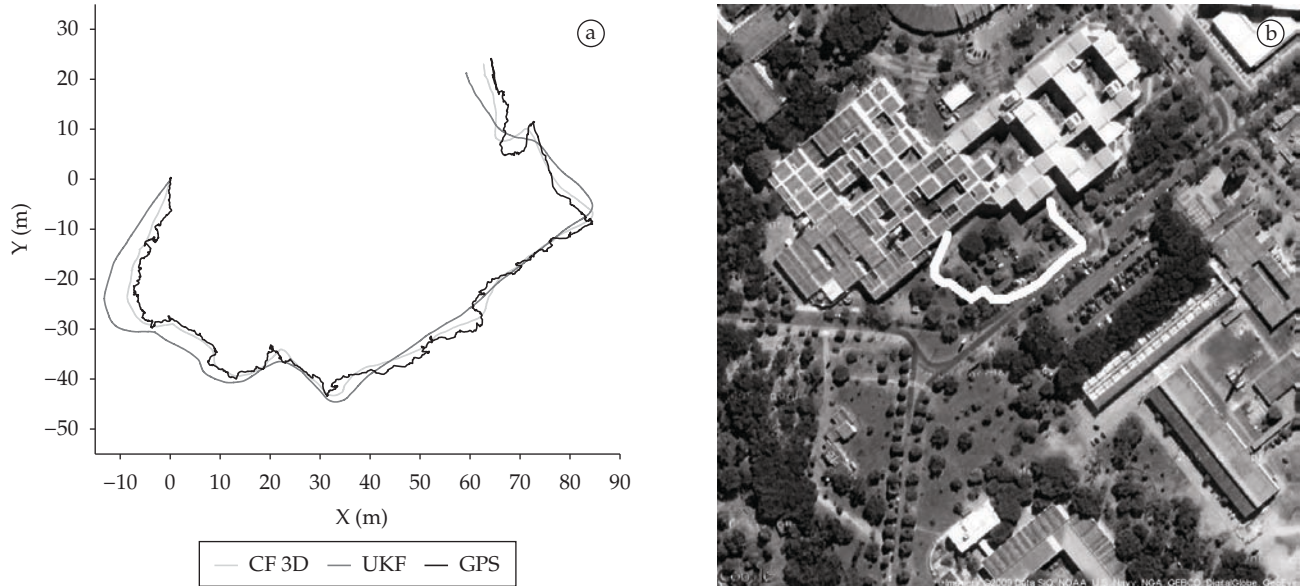


Figure 9. Reconstruction of the path performed by the robot in Experiment 2.

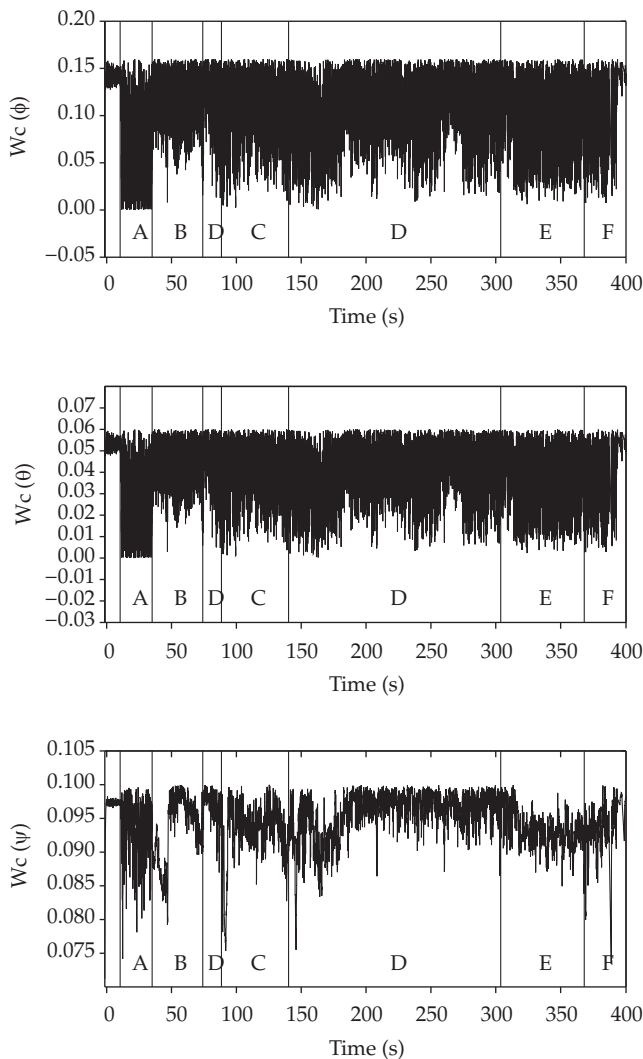


Figure 10. Calculated ω_c for the orientation angles during Experiment 2.

Figure 12a shows how the speed of the robot varied throughout the trajectory. It is notable that lower speeds occur mainly during at curves, which some times are made in a turn-in-place manner due to the fact that the robot's is of the skid-steer type. Figure 12b show the ω_c calculated based on the velocity of the robot.

Figure 13 shows the results for the simulation of signal loss. It can be observed that a small error on the orientation generates large errors in the localization, where much of the error is related to the odometry. However, once the signal of the GPS becomes available again the path gets corrected.

5. Conclusions and Future Works

This work presented the use of the CF technique to estimate the full pose (localization and orientation) in the configuration space, for mobile robots navigating in outdoor environments on uneven terrain. The main advantage of the proposed technique is the significant reduction of the computational complexity of the overall system, which enables increasing the sampling frequency of the signals with a consequent improvement in the accuracy of the estimates.

The results were satisfactory and very close to the values obtained with more complex techniques such as the UKF. This has been shown by plotting the actual path on satellite images, where the estimated path performed by using the proposed methodology have shown to be faithful to the actual path performed by the robot. We have also shown that height (change in Z axis) estimation was less accurate, mainly due to the large uncertainty on the altitude data provided by the GPS.

It was possible to obtain a good estimation of the orientation angles (ϕ , θ , ψ), but they still have a certain amount of noise. The most accurate and reliable value was ψ , mainly due to the use of a digital compass, which made it possible to eliminate drift.

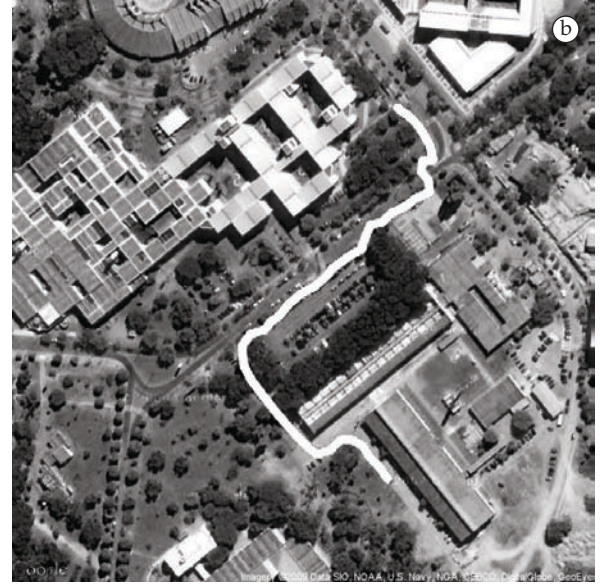
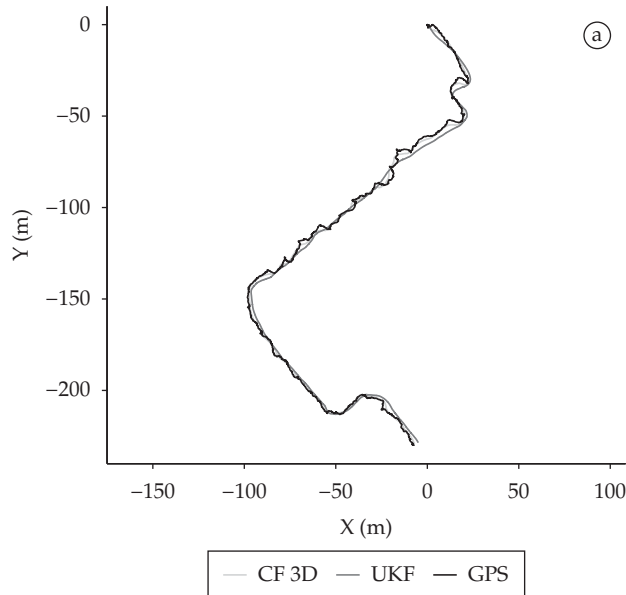


Figure 11. Reconstruction of the path performed by the robot in Experiment 3.

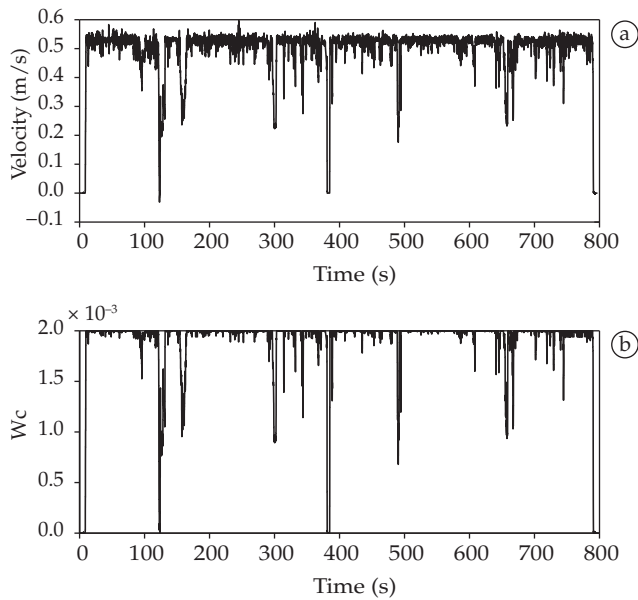


Figure 12. Comparison between the calculated ω_c (b) and the robot speed (a).

Although the experiments were conducted in an offline way (post navigation processing), it was possible to observe that the method is very efficient, and amenable to be used in real time.

As future work, further experiments will be performed with larger vehicles, since it will make it possible to analyze the behavior of the system on greater scale experiments where the vehicle's size is comparable to the path width restrictions.

Sensor processing continues to be an area of great relevance for robotics. Further investigation is demanded on sensors imperfections, such as bias in the IMU accelerometers

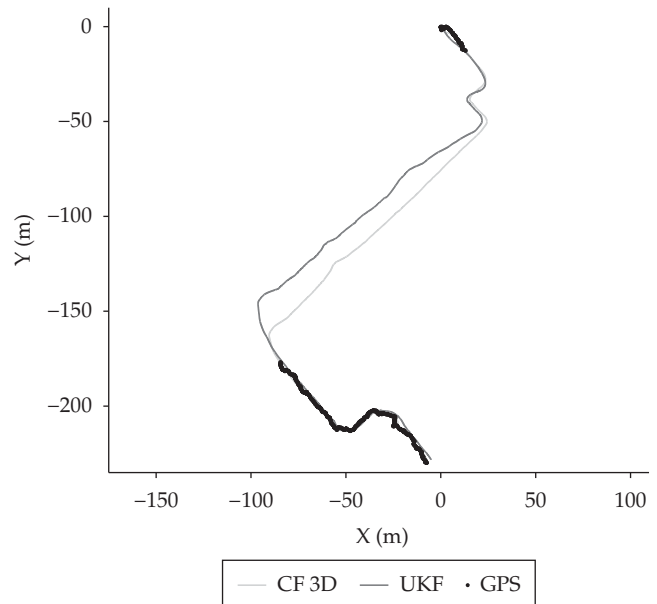


Figure 13. Reconstruction of the path executed by the robot in Experiment 3 with GPS signal loss (simulated).

and gyros, in order to provide better methods to minimize even further their effects on the final estimations. Likewise, issues arising in other sensors such as magnetometers must be dealt with adequately.

Finally, the estimation of the variation in the altitude of the robot is very problematic, and it can be the source of a large amount of errors in the system. To tackle this problem, it is necessary to perform better analysis and treatment of the GPS altitude data. Other sensors, such as accurate differential pressures sensors, may be considered as an alternative to GPS altitude data.

Acknowledgements

The authors would like to thank Wolmar Pimenta for the support on the construction and integration of the mechanical system used in the experiments, and the financial support offered by the Fundação de Amparo a Pesquisa do Estado de Minas Gerais (FAPEMIG), the Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES).

References

1. Baerveldt AJ and Klang R. A low-cost and low-weight attitude estimation system for a autonomous helicopter. In: *Proceedings of IEEE International Conference on Intelligent Engineering Systems*; 1997; Budapest, Hungary. IEEE; 1997.
2. Coppersmith D and Winograd S. Matrix multiplication via arithmetic progressions. In: *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*; 1987; New York. ACM; 1987. p. 1-6.
3. Euston M, Coote P, Mahony R, Kim J and Hamel T. A complementary filter for attitude estimation of a fixed-wing uav. In: *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*; 2008; Nice, France. IEEE; 2008. p. 340-345.
4. Google Maps API. *Google Code*. [on the internet] 2008. Available from: <http://code.google.com/apis/maps/>. Access in: 29/07/2008.
5. Günthner W. *Enhancing cognitive assistance systems with inertial measurement units*. New York: Springer; 2008.
6. Iscold P, Oliveira GRC, Neto AA, Pereira GAS and Torres LAB. Desenvolvimento de horizonte artificial para aviação geral baseado em sensores MEMS. In: *Anais do 5 Congresso Brasileiro de Engenharia Inercial*; 2007; Rio de Janeiro.
7. Jeffrey SJ and Uhlmann K. A new extension of the kalman filter to nonlinear systems. In: *Proceedings of the 11 International Symposium on Aerospace/Defense Sensing, Simulation and Controls*; 1997; Orlando, Florida. International Society for Optical Engineering; 1997.
8. Li Y, Wang J, Rizos C, Mumford P and Ding W. Low-cost tightly coupled gps/ins integration based on a nonlinear kalman filtering design. In: *Proceedings of the National Technical Meeting of the Institute of Navigation*; 2006; San Diego. p. 958-966.
9. Liu B, Adams M and Ibañez-Guzmán J. Multi-aided inertial navigation for ground vehicles in outdoor uneven environments. In: *Proceedings of IEEE International Conference on Robotics and Automation*; 2005; Barcelona, Espanha. IEEE; 2005.
10. Mahony R and Hamel T. *Advances in unmanned aerial vehicles: state of the art and the road to autonomy, chapter robust nonlinear observers for attitude estimation of mini uavs*. New York: Springer; 2007. p. 343-376.
11. Müller M, Surmann H, Pervözl K and May S. The accuracy of 6D SLAM using the AIS 3D laser scanner. In: *Proceedings of International Conference on Multisensor Fusion and Integration for Intelligent Systems*; 2006; Heidelberg, Germany.
12. Ojeda L and Borenstein J. *Improved position estimation for mobile robots on rough terrain using attitude information*. Michigan: The University of Michigan; 2001. (Technical report)
13. Sukkarieh S, Nebot EM and Durrant-Whyte HF. A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. *IEEE Transactions on Robotics and Automation*. 1999; 15(6):572-578.
14. Thrun S, Burgard W and Fox D. *Probabilistic robotics: intelligent robotics and autonomous agents*. Cambridge, MA: The MIT Press; 2005.
15. Thrun S, Diel M, and Hähnel D. Scan alignment and 3D surface modeling with a helicopter platform. In: *Proceedings of the International Conference on Field and Service Robotics*; 2003; Lake Yamanaka, Japan.
16. Walchko KJ and Mason PAC. *Inertial Navigation*. In: *Proceedings of Florida Conference on Recent Advances in Robotics*; 2002; Miami, Florida.